PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : H04L 29/06, 29/12 | A1

(11) International Publication Number: **WO 00/19682**

(43) International Publication Date: 6 April 2000 (06.04.00)

(21) International Application Number: PCT/US99/20720

(22) International Filing Date: 10 September 1999 (10.09.99)

(30) Priority Data:
09/161,326 25 September 1998 (25.09.98) US

(71) Applicant: SUN MICROSYSTEMS, INC. [US/US]; 901 San Antonio Road, MS PAL1-521, Palo Alto, CA 94303 (US).

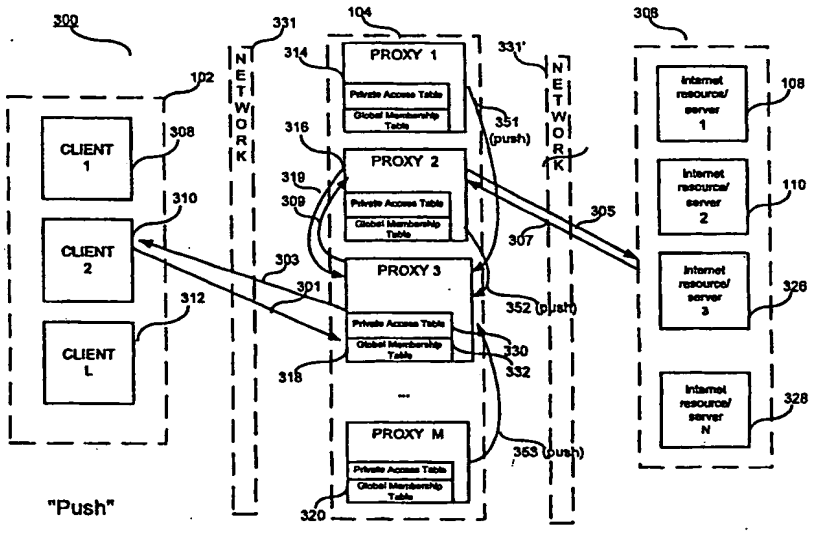
(72) Inventor: GUPTA, Amit; 2000 Walnut Street, Apartment J-207, Fremont, CA 94538 (US).

(74) Agents: SOCKOL, Marc, A.; Graham & James LLP, 600 Hansen Way, Palo Alto, CA 94304-1043 (US) et al.

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: AN APPARATUS AND METHOD FOR IMPROVING PERFORMANCE OF PROXY ARRAYS THAT USE PERSISTENT CONNECTIONS



(57) Abstract

A method and apparatus for improving the performance of a proxy array to access and retrieve information from a server. Specifically, when a proxy receives a request for a resource, such as a Web page, the proxy first determines it has established a persistent connection with the server on which the Web page resides. If the receiving proxy does not have a persistent connection, if another proxy has established a persistent connection to that server. If there is an established connection, the proxy that received the request instructs the other proxy to retrieve the requested data from the server and transfer the data to the client. Otherwise, if there is no existing persistent connection, the proxy creates a persistent connection to the desired server.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

An Apparatus and Method for Improving Performance of Proxy Arrays That Use Persistent Connections

FIELD OF THE INVENTION

This invention relates generally to client-server computer networks. More particularly, this invention relates to a general technique for improving the performance of a proxy array during persistent network access.

BACKGROUND OF THE INVENTION

The recent publicity and emphasis on the information superhighway has increased the awareness and acceptance of the Internet as a mass communications medium. Until recently, "cruising or surfing" the Internet was a disorienting, even a frustrating experience, something like trying to navigate without maps. The World Wide Web has made it easier to access the array of resources available on the Internet. Resources, such as web servers, ftp servers, and telnet servers, provide the user with the ability to easily find the data content or information he wants simply and easily.

The volume of World Wide Web traffic on the Internet is staggering but a significant fraction of this traffic is redundant. That is, a large number of users request the same data from the same resource, at around the same time. As a result, a significant percentage of a corporation's network infrastructure is carrying and servicing the repeated requests for same data content, day after day.

To manage this growing demand for access to the Internet and to reduce network communications costs, some networks include "proxy arrays (also called "proxy server arrays"), which are made up of one or more "proxies" (also called "proxy servers") . Proxies are network server-based applications that are placed between a client application, such as a web browser, and a resource, such as a Web server. Initially proxies were designed to deal with problems caused by firewall issues in corporate web access. Eventually, proxies were also recognized as being an ideal environment to cache web data and to improve system performance, as well as to reduce the load on the network and on the servers.

In most World Wide Web based client/server applications, the proxy receives a request to access a specific resource from a client system. The proxy sends a request to the desired resource or site specified by the Uniform Resource Locator (URL). The URL acts as the address of the resource and as such is unique throughout the Internet. The proxy retrieves the Web page from the resource specified by the URL address and transfers the Web page to the client. (Alternately, some proxies may already have the page stored in their cache and do not need to retrieve it from the specified resource). If the proxy has a cache, the proxy also stores the retrieved Web page in its cache for future use.

It is becoming common for networks (such as intranets or the Internet) to include a plurality of proxies, each accessible by multiple clients. Requests from clients for various pages stored on servers in the network are routed through the plurality of proxies, which cache pages whenever possible.

Certain systems allow the client to specify to which proxy it will send its request. Thus, one client may send requests to more than one proxy. While proxies improve the overall performance of a network, having more than one proxy access the same Web page is inefficient. For example, for proxies that include a cache, allowing multiple proxies to retrieve the same Web page will result in more than one proxy storing the same Web page in its cache.

The World Wide Web operates using the http protocol. An initial version of http (http 1.0) required a separate TCP connection for each transfer of information between a client and a proxy or a server. Subsequently, newer versions of the http protocol have reduced the need to establish a separate connection for each request. For example, version 1.1 of the http protocol includes "persistent http," where multiple http transfers can use the same connection, i.e., the same proxy without having to establish a new connection each time. Persistent http connections are described in, for example, Request For Comments (RFC) 2068, "Hypertext Transfer Protocol -- HTTP/1.1," January 1997, available from the Internet Engineering Taskforce (IETF).

A problem arises, however, if a client can send its requests to more than one proxy, since clients generally do not know whether a proxy has a persistent connection to the web server or not. Furthermore, if one or more clients can access

one or more proxies, the clients do not know which of the proxies have already received requests for a particular service from another client (and may therefore have a persistent connection open).

In some conventional systems, the Internet Cache Protocol (ICP) is used to determine and select the most applicable location from which to retrieve a requested Web page. In ICP, one proxy establishes a "working" relationship with other proxies. Proxies designated as parents are on one level while child proxies are on lower level(s). The terms "neighbor" and "peer" refer to either a parent or a sibling that are a single "cache reference" away.

In general, in ICP the flow of a client request is up through the hierarchy of proxies. If a proxy does not have a client's requested Web page, it requests that a special proxy, called an arbitrator, query the other proxies to see if they have the desired Web page. If any of these proxies has the requested Web page in its cache, then the inquiring proxy enters a demand for the Web page. The cached Web page is either forwarded directly to the client or to the original proxy for transfer to the client. If none of the proxies have the Web page in their cache, the proxy must forward the request either to a parent or back to the original Web for service. Thus, if a successful request or "hit" occurs, it may fetch the Web page from a peer proxy or the requested wed page is received from a parent but if the request is unsuccessful or "missed," it must be passed to a parent server for service. The role of a parent is to complete the transaction and to service the request. If necessary, a parent proxy will connect directly to a resource (such as a web service) directly to service a client's request.

There are several problems that arise with the ICP approach. For example, the arbitrating proxy may be overrun with requests or the network path between proxies may be congested. In addition, the additional hierarchy introduces extra delays for the clients requesting uncached data.

Other conventional client/server systems, such as the CARP (Cache Array Routing Protocol), available from Microsoft, Inc. of Redmond, Washington access a variety of proxies to retrieve pages stored on a single server. CARP clients, for example, use a deterministic hashing function to allocate page accessing and caching

among a variety of proxies. By accessing a variety of proxies for various pages stored on a server, the CARP system aims to achieve load balancing between the proxies. Unfortunately, such a system has the disadvantage that it requires a hashing function that distributes the page accesses equally among the proxies. If the hashing function is poorly chosen or if the URL names lead to unbalanced distribution of the URLs to various proxies, it will negatively affect the load balancing between proxies. Thus, CARP's use of a deterministic hashing function to distribute requests to proxies does not always achieve a good distribution of requests among proxies. Lack of good distribution leads to inefficient usage of proxies. More importantly, with CARP, two URLs for the same Web server are likely to be sent to two different proxies, thereby undoing the benefits of persistent connections between a particular proxy and a server.

SUMMARY OF THE INVENTION

An embodiment of the present invention provides a method and apparatus for improving the performance of a proxy array (also called a "proxy server array") using a persistent connection to access and retrieve information from a resource provider (also called a "server"). Specifically, when a proxy receives a request for a network resource, such as a Web page, the proxy first determines whether it has a persistent connection to the server on which the resource resides. If so, the proxy retrieves the resource via the persistent connection. If not, the proxy determines whether another proxy has established a persistent connection to that server. If there is an existing persistent connection, the proxy that received the request instructs the other proxy with the persistent connection to retrieve the requested data from the server. Otherwise, if there is no existing persistent connection, the proxy that received the request creates a persistent connection for the desired server.

At least two embodiments of the invention are described herein. A first embodiment is a "push" embodiment in which information about persistent connections are exchanged ("pushed") among proxies ahead of time, so that the information about existing persistent connections of all proxies will be available to a particular proxy when a request is received from a client. A second embodiment is a

"pull" embodiment in which information about persistent connections is exchanged ("pulled") among proxies in response to a request received from a client.

In accordance with the purpose of the invention, as embodied and broadly described herein, the invention relates to a method of accessing information in a network, comprising the steps, performed by at least one proxy in the network, of: receiving a request for access to a network resource stored on a server, wherein the request includes a network resource address corresponding to the network resource; determining an address of the server from the network resource address; comparing the server address against a plurality of server addresses stored in a first table of the proxy, wherein each stored address corresponds to a server to which the proxy has a persistent connection; if a match for the server address cannot be found in the first table, interrogating a second table of the proxy, which contains a list of at least one other proxy and corresponding stored addresses to which respective ones of the other proxies have a persistent connection, to find an address that matches the server address; and transferring the request for access, if an address stored in the second table matches the server address, to another proxy that corresponds to the address found in the second table.

In further accordance with the purpose of the invention, as embodied and broadly described herein, the invention relates to a method of accessing information in a network, comprising the steps, performed by at least one proxy in the network, of: receiving a request for access to a network resource stored on a server, wherein the request includes a network resource address corresponding to the network resource; determining an address of the server from the network resource address; comparing the server address against a plurality of server addresses stored in a first table of the proxy, wherein each stored address corresponds to a server to which the proxy has a persistent connection; broadcasting, if a match for the server address cannot be found in the first table, a request for service to other proxies listed in a second table of the proxy; receiving an indication from at least one other proxy that a persistent connection exists between the other proxy and the server; and transferring, if a persistent connection exists between the other proxy and the server, the request for access to the other proxy.

Advantages of the invention will be set forth, in part, in the description that follows and, in part, will be understood by those skilled in the art from the description herein. The advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims and equivalents.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate several embodiments of the invention and, together with the description, serve to explain the principles of the invention.

Fig. 1 is a block diagram of a typical client/server network on which an embodiment of the present invention can be implemented.

Fig. 2 is a block diagram of a data processing system in accordance with an embodiment of the invention.

Figs. 3(a) and 3(b) are block diagrams showing a network in accordance with a "push" embodiment of the present invention.

Fig. 4(a) is a flow chart showing steps performed by a receiving proxy in response to a client request for access to a server in accordance with the embodiment of Figs. 3(a) and 3(b).

Fig. 4(b) is a flow chart showing steps performed by another proxy in response to a message from the receiving proxy to access a server in accordance with the embodiment of Fig. 3(a).

Fig. 4(c) is a flow chart showing steps performed by another proxy in response to a message from the receiving proxy to access a server in accordance with the embodiment of Fig. 3(b).

Fig. 4(d) is a flow chart showing steps performed to push information about existing persistent connections to other proxies.

Figs. 5(a) and 5(b) are block diagrams showing a network in accordance with a "pull" embodiment of the present invention.

Fig. 6(a) is a flow chart showing steps performed by a proxy in response to a client request for access to a server in accordance with Fig. 5.

Fig. 6(b) is a flow chart showing steps performed by a proxy in response to a request for persistent connection information from another proxy.

Fig. 7 shows an exemplary format of a private access table in a proxy in the embodiments of Figs. 3 and 5.

Fig. 8 shows an exemplary format of a global access table in a proxy in the embodiment of Fig. 3.

Fig. 9 shows an exemplary format of a proxy support table in a proxy in the embodiment of Fig. 5.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The present invention now will be described more fully with reference to the accompanying drawings, in which several embodiments of the invention are shown. The present invention may, however, embodied in many different forms and should not be construed as limited to the embodiments set forth herein, rather these embodiments are provided so that this disclosure will be thorough and complete and will fully convey the invention to those skilled in the art.

Fig. 1 is a block diagram illustrating a typical client/server computer network 100, in accordance with the present invention. The computer network 100 includes a client site 102 coupled via a network interface or a connection 101 and network 103 to a proxy site 105 (such as a proxy array) having proxy servers (or "proxies")104. Proxies 104 are able to communicate with one or more resources/servers 108, 110 via a connection 111 and a network 106. Typically, a client site includes several client systems 112.

Fig. 2 shows a data processing system 200 that is programmed to perform the functions of a system, such as one of the proxies 104. The system 200 includes a processor 202 (or any appropriate processor or processors) and some form of storage 204. A portion of the storage 204 contains the software and data of the present invention. Storage 204 is capable of storing system software 218, and a plurality of access tables 220. The access tables 220 include various tables of Figs. 7-9, depending on the embodiment implemented. Certain embodiments also include a

cache 222 containing a plurality of Web pages, although a cache is not required in all implementations of the present invention.

System 200 preferably also includes an input device 208, such as a keyboard, mouse, touch screen, voice control etc. System 200 preferably also includes an output device 210, such as a printer, display screen, or voice output device. System 200 preferably also includes a computer readable medium input device 214 that inputs instructions and/or data from a computer readable medium 212. The term "computer-readable medium" as used herein refers to any medium that participates in providing instructions to a processor for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. The instructions can also be transmitted via a carrier wave in network 101, 111, such as a LAN, a WAN, or the Internet.

In the following discussion, it is understood that the appropriate processor 202 (or similar processors) perform the steps of methods and flowcharts discussed preferably executing instructions stored in storage 204. It will also be understood that the invention is not limited to any particular implementation or programming technique and that the invention may be implemented using any appropriate techniques for implementing the functionality described herein. The invention is not limited to any particular programming language or operating system. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the present invention are not limited to any specific combination of hardware circuitry and software.

The above paragraphs describe a proxy array 104. Client system 112 also include a processor and some form of storage (not shown). The storage element of the client system 112 stores system software, computer programs, platform-independent binaries produced by compilers, and software, such as a web browser to communicate with a server via one of the proxies 104.

Figs. 3(a) and 3(b) are block diagrams 300, 300' of two different versions of a "push" embodiment, showing clients 102, proxies 104, and a plurality of servers 306. In a "push embodiment," each proxy pushes information about its persistent connections to the other proxies. Each proxy 104 has a private access table, which

indicates the servers with which the respective proxy currently has a persistent connection, and a global membership table, which indicates which other proxy tables have a persistent connection. In the described embodiment, the global membership tables are the same in each proxy and are updated via the pushed information.

In the "push" embodiment of Fig. 3(a), as described below in detail, all proxies 104 "push" information 351, 352, 353 about their persistent connections to the other proxies. Fig. 3(a) shows a proxy 318 receiving a request 301 for a resource, such as a Web page, stored on a server 326 from a client system 310 (step 402 of Fig. 4). The receiving proxy itself will retrieve the page, if it has an existing persistent connection with the server 326 on which the page resides. If another proxy has a persistent connection with that server, as indicated in the global access table of the receiving proxy, that other proxy will be instructed 309 to retrieve the page. Otherwise, receiving proxy 318 opens a new persistent connection, retrieves the Web page, sends the page to the requesting client, and updates the tables to reflect the new persistent connection (not shown).

As shown in Fig. 3(a), if another proxy 316 has an existing persistent connection to server 326, it receives instruction 309 from the receiving proxy, retrieves the Web page 305, 307, and sends 319 the retrieved Web page to the receiving proxy, which sends 303 the Web page to the client.

Fig. 3(b) is similar to Fig. 3(a), except that, if another proxy 316 has an existing persistent connection to server 326 , it receives instruction 309 from the receiving proxy, retrieves the Web page 305, 307, and sends 303 the Web page directly to the client, without going through the receiving proxy.

Fig. 4(a) is a flowchart illustrating the steps performed by a proxy to service a client request in accordance with Figs. 3(a) and 3(b). As shown in the example of Fig. 4(a), a proxy (such as proxy 318) in step 402 receives a request 301 for a Web page from a client, such as client 310. In the example, the Web page is stored on server 326. The proxy receiving the request is called "the receiving proxy." The client 310 initially determines to which proxy to send the request using any appropriate method. The request includes an address of a Web page that is stored on server 326. If receiving proxy 318 has a cache, receiving proxy 318 in step 406 searches its cache

for the Web page that corresponds to the client's requested page and determines if the requested Web page is in its cache. If the Web page is present in local cache, receiving proxy 318 in step 408 retrieves the requested Web page from its cache in step 408 and transmits the requested Web page to client 310.

If the Web page is not in local cache (or the proxy does not have a cache), the receiving proxy in step 410 truncates the Web page address in the request to yield the address of the Web server that stores the requested page. Thus, for example, if the address is URL = http://www.companyA.com/page1, the truncated address is URL = http://www.companyA.com. In step 412, receiving proxy 318 searches its private access table 330 for the truncated address. As shown in the exemplary format of Fig. 7, the private access table indicates the servers for which receiving proxy 318 has an open persistent connection. The address is truncated because a proxy can have a persistent connection with a server even if it has not previously fetched all Web pages residing on the server. The proxy can, for example, have previously fetched a different Web page from the server, and can thus have a persistent connection open even if it has not previously retrieved the requested Web page. Thus, it is necessary to check whether a persistent connection exists for a Web server, not for a particular page.

Receiving proxy 318, in step 412, determines if the truncated address is present in private access table 330 of receiving proxy 318. If the truncated address is present (step 414), the proxy in step 422 completes the transaction by making use of its existing persistent connection with the server to retrieve 305, 307 the requested Web page from server 326. Thus, receiving proxy 318 can take advantage of its existing persistent connection. Note that the entire address of the Web page, not the truncated address, is sent to the server 326 to retrieve the Web page. Upon receipt of the requested Web page from the server, the receiving proxy in step 422 sends a copy of the Web page to requesting client 310. Receiving proxy 318 also stores a copy of the Web page in cache (if the proxy includes a cache).

If the truncated address cannot be located in private access table 330 of receiving proxy 318, either some other proxy has a persistent connection to the server or no proxy has a persistent connection to the server. In step 416, the receiving proxy

searches for the requested truncated address in a second table stored in its storage 204 called the global membership table 332. As shown in Fig. 8, the global membership table contains a list of all the other proxies 104 and the servers to which the other proxies have persistent connections (along with the associated ports for the connections).

The proxy determines, in step 416, if the truncated requested address is in global membership table 332 of receiving proxy 318. If the truncated address is found, the proxy in step 420 transfers 309 the request for the Web page to the appropriate proxy found in the global membership table 332. In the example, this other proxy is proxy 316.

Fig. 4(b) shows an implementation of retrieving and transferring the retrieved web page to the requesting client shown in Fig. 3(a). In response to the transferred request 309, the other proxy 316 completes the transaction by retrieving the requested Web page 305, 307 over its open persistent connection to server 326. The other proxy in step 494 retrieves and transfers the requested Web page to the receiving proxy, which sends it to the requesting client (step 496).

Fig. 4(c) shows an implementation of retrieving and transferring the retrieved web page to the requesting client shown in Fig. 3(b). In response to the transferred request 309, the other proxy 316 completes the transaction by retrieving the requested Web page 305, 307 over its open persistent connection to server 326. The proxy in step 456 then transfers the requested Web page to the client. Note that this step requires the proxy to send the Web page to the client with headers "pretending" that the Web page is being sent by the receiving proxy.

If, in Fig. 4(a), the truncated requested address is not in the global membership table of the receiving proxy, no persistent connections are open for the server storing the requested page. In this case, in step 418, the receiving proxy that originally received the client request completes the transaction by initiating a persistent connection to the server 326. Upon completion, the receiving proxy in step 422 retrieves and transfers the requested Web page to the client, stores a copy of the Web page in its cache (if it includes a cache), and updates its private access table .

The receiving proxy also sends a message to the other proxies to update their global access tables to reflect the new persistent connection.

In the example of Fig. 3(a), client 310 sends a request 301 for a Web page to proxy 318. In accordance with the flow chart of Fig. 4(a), proxy 318 determines that the Web page is stored on server 326 and that proxy 316 has an open persistent connection to the server. Proxy 316 sends a request 305 to server 326. One of the other proxies (or the receiving proxy) then sends the Web page 307 to the requesting client 310 (see Fig. 4(b) or Fig. 4(c)).

Fig. 4(d) is a flow chart showing steps to push information about persistent connections to other proxies. In one embodiment, this push step 452 is performed periodically. In another embodiment, this push step 452 is performed after a predetermined number of changes have occurred (e.g., after three new connections and/or termination of existing persistent connections). In yet another embodiment, push step 452 is performed after some combination of number of changes and time passed since the first or last change. In certain embodiments, the push step may be performed at different times and in accordance with different criteria in different proxies. Each proxy sends a message or messages 351, 352, 353 to other proxies about the persistent connections that it currently has open. As discussed above, this information is used to update the global access tables in each proxy.

Fig. 5 is a block diagram 500 showing clients 102, proxies 104, and a plurality of servers 506. Each proxy has a private access table, which indicates the servers with which that proxy has a persistent connection, and a proxy support table, which indicates the other proxy tables that exist. In the described embodiment, the proxy support tables are the same in each proxy. In the described embodiment, each table entry can be one or more unicast or a multicast addresses. Thus, one or more other proxies can have an existing persistent connection to the server.

In the "pull" embodiment of Fig. 5(a), as described below in detail, a proxy 518 receives a request for a resource, such as a Web page, stored on a server 524 from a client system 510 (step 602 of Fig. 6(a)). The receiving proxy itself will retrieve the page, if it has an existing persistent connection with the server on which the page resides. If no such persistent connection exists, the receiving proxy broadcasts 551,

552, 553 a request for proxies with persistent connections to the server 524. If one or more other proxies have a persistent connection with that server, the other proxy or proxies will inform 554 the receiving proxy, which informs 555 one of the responding proxies, for example, proxy 1 514, to retrieve the page and transfer 556 it back to the receiving proxy. Otherwise, proxy 518 opens a new persistent connection (not shown), retrieves the Web page, and sends the page to the requesting client.

Fig. 5(b) is similar to the "push" embodiment of Fig. 5(a) except that the other proxy 514 will itself send 503 the retrieved Web page back to the requesting client 510.

Figs. 6(a) and 6(b) are flow charts, in accordance with Figs. 5(a) and 5(b), that show the steps of a broadcast technique used by the present invention to obtain a desired Web page from a server while making use of existing persistent connections. In step 602, a proxy 518 receives a request for access to a Web page stored on a server 524 from a client 510.

The proxy in step 606 determines if the requested Web page resides in its cache (if the proxy includes a cache). If the page is present in cache, the proxy in step 608 retrieves and transmits the requested Web page to the client. If the Web page is not in the cache (or the proxy does not include a cache), the proxy in step 610 truncates the address of the Web page to yield the address of the server on which the page resides. For example, if the Web page address is "http://companyA.com/page1" the truncated address (i.e., the address of the server on which the page resides) is "http://companyA.com".

If in step 612, the receiving proxy determines that the truncated address is present in the private access table, the receiving proxy has an existing persistent connection with the server on which the Web page resides. The receiving proxy then completes the transaction by retrieving the Web page from the server via the persistent connection. In step 622, upon receipt and transfer of the requested Web page to the client, the receiving proxy stores a copy of the Web page in an appropriate location in cache for future use (if the proxy includes a cache). If the truncated address is not in the private access table of the receiving proxy, the receiving proxy, in step 613, broadcasts (e.g., unicasts and/or multicasts) a service

request or inquiry 551, 552, 553 to the other proxies listed in the proxy support table 532.

An exemplary format of the private access table has been discussed above. An exemplary format of the proxy support table is shown in Fig. 9. The proxy support table contains a list of all proxies that might have a persistent connection to a server. In the described embodiment, an entry in the proxy support table can be either multicast address or unicast addresses. The addresses in the proxy support table are the addresses to which the proxy will broadcast its service request (i.e., the addresses of all other proxies or a sub-set of all other proxies).

The receiving proxy waits for a "pulled" response 554 to its broadcast from at least one other proxy. If no response is received in a predetermined period of time, a timeout occurs and control passes to step 618. If the receiving proxy receives a positive response in step 616, the receiving proxy in step 620 transfers 555 the request to one of the responding proxies. This proxy has an existing persistent connection to the server on which the Web page resides. If more than one proxy responds to the service request 551, 552, 553, the receiving proxy chooses between the responding proxies using any appropriate method, such as at random, round robin, etc.

As shown in Figs. 4(b) and 4(c), which were discussed above, responsive to the transferred request 555, the responding proxy completes the transaction by making use of its existing persistent connection to the appropriate server and retrieving the requested Web page. The proxy sends the retrieved Web page to either the requesting client (see Fig. 4(c)) or to the receiving proxy (see Fig. 4(b)).

Returning to step 618 of Fig. 6(a), if the receiving proxy does not receive a response to its inquiry 551, 552, 553 (or a time out occurs), the receiving proxy in step 618 opens a new persistent connection between the receiving proxy and the server upon which the Web page resides. Upon establishment of the persistent connection, the proxy in step 622 retrieves and transfers the Web page to the client, stores a copy of the Web page in its cache (if the receiving proxy includes a cache) , and updates its private access table to reflect that the persistent connection exists.

Fig. 6(b) shows an example of steps performed by a proxy when it receives a "pull" request from a receiving proxy. In step 654, the proxy determines whether a truncated version of the address is in the proxy's private access table, If so, the proxy determines whether its load exceeds a configured limit. If the load exceeds the configured limit, the proxy is too busy to make use of its persistent connection, even though it has one. Step 656 is optional and may not be implemented in all embodiments. In step 658, the proxy sends information about its persistent connections to the receiving proxy, in response to the request for information and control returns to step 616 of Fig. 6(a).

While we have described embodiments of the present invention, it is understood that those skilled in the art, both now and in the future, may make various improvements and enhancements which fall within the scope of the claims which follow. These claims should be construed to maintain the proper protection for the invention disclosed.

WHAT IS CLAIMED IS:

1. A method of accessing information in a network, comprising the steps, performed by at least one proxy in the network, of:

receiving a request for access to a network resource stored on a server, wherein the request includes a network resource address corresponding to the network resource;

determining an address of the server from the network resource address;

comparing the server address against a plurality of server addresses stored in a first table of the proxy, wherein each stored address corresponds to a server to which the proxy has a persistent connection;

if a match for the server address cannot be found in the first table, interrogating a second table of the proxy, which contains a list of at least one other proxy and corresponding stored addresses to which respective ones of the other proxies have a persistent connection, to find an address that matches the server address; and

transferring the request for access, if an address stored in the second table matches the server address, to another proxy that corresponds to the address found in the second table.

2. The method as recited in claim 1, further comprising retrieving, if the server address corresponds to an address stored in the first table of the proxy, by the proxy, information from the server at that address via an existing persistent connection between the proxy and the server.

3. The method as recited in claim 2, further comprising sending, by the proxy, the information retrieved from the server to the client.

4. The method as recited in claim 1, further comprising updating, when a persistent connection is terminated by the proxy, the first and second tables to indicate that the persistent connection no longer exists.

5. The method as recited in claim 1, further comprising initiating, if the second table does not contain a matching address for the server address, initiating, by the proxy that originally received the request, a persistent connection to the server and retrieving the requested information.

6. The method as recited in claim 5, further comprising forwarding by the proxy the retrieved information to the client and updating the first and second tables with the server address used and a corresponding proxy identifier of the proxy that serviced the request.

7. The method as recited in claim 1, further comprising retrieving by the other proxy the requested information, sending the retrieved information to the proxy and forwarding by the proxy the retrieved information to the client.

8. The method as recited in claim 1, further comprising retrieving by the other proxy the requested information and forwarding by the other proxy the retrieved information to the client.

9. A method of accessing information in a network, comprising the steps, performed by at least one proxy in the network, of:

receiving a request for access to a network resource stored on a server, wherein the request includes a network resource address corresponding to the network resource;

determining an address of the server from the network resource address;

comparing the server address against a plurality of server addresses stored in a first table of the proxy, wherein each stored address corresponds to a server to which the proxy has a persistent connection;

broadcasting, if a match for the server address cannot be found in the first table, a request for service to other proxies listed in a second table of the proxy;

receiving an indication from at least one other proxy that a persistent connection exists between the other proxy and the server; and

transferring, if a persistent connection exists between the other proxy and the server, the request for access to the other proxy.

10. The method as recited in claim 9, further comprising retrieving, if the server address corresponds to an address stored in the first table of the proxy, by the proxy, information from the server at that address via an existing persistent connection between the proxy and the server.

11. The method as recited in claim 10, further comprising sending, by the proxy, the information retrieved from the server to the client.

12. The method as recited in claim 11, further comprising when a persistent connection is terminated, updating, by the proxy, the first and second tables to indicate that the persistent connection no longer exists.

13. The method as recited by claim 9, further comprising, if the broadcast request does not yield a response from any of the listed proxies and a timeout occurs, initiating, by the proxy that originated the broadcast request, a persistent connection to the server and retrieving the requested information.

14. The method as recited in claim 13, further comprising forwarding by the proxy the retrieved information to the client.

15. The method as recited in claim 9, further comprising retrieving by the other proxy the requested information, sending the retrieved information to the proxy, and forwarding by the proxy the retrieved information to the client.

16. The method as recited in claim 9, further comprising retrieving by the other proxy the requested information and forwarding by the other proxy the retrieved information to the client.

17. An apparatus that accesses information in a network, comprising:

a portion of a proxy configured to receive a request for access to a network resource stored on a server, wherein the request includes a network resource address corresponding to the network resource;

a portion of the proxy configured to determine an address of the server from the network resource address;

a portion of the proxy configured to compare the server address against a plurality of server addresses stored in a first table of the proxy, wherein each stored address corresponds to a server to which the proxy has a persistent connection;

a portion of the proxy configured to, if a match for the server address cannot be found in the first table, interrogate a second table of the proxy, which contains a list of at least one other proxy and corresponding stored addresses to which respective ones of the other proxies have a persistent connection, to find an address that matches the server address; and

a portion of a proxy configured to transfer the request for access, if an address stored in the second table matches the server address, to another proxy that corresponds to the address found in the second table.

18. The apparatus as recited in claim 17, further comprising a portion of the proxy configured to retrieve, if the server address corresponds to an address stored in the first table of the proxy, information from the server at that address via an existing persistent connection between the proxy and the server.

19. The apparatus as recited in claim 18, further comprising a portion of the proxy configured to send the information retrieved from the server to the client.

20. The apparatus as recited in claim 17, further comprising a portion of the proxy configured to update, when a persistent connection is terminated by the proxy, the first and second tables to indicate that the persistent connection no longer exists.

21. The apparatus as recited in claim 17, further comprising a portion of the proxy configured to initiate, if the second table does not contain a matching address for the server address, a persistent connection to the server and to retrieve the requested information.

22. The apparatus as recited in claim 21, further comprising a portion of the proxy configured to forward the retrieved information to the client and to update the first and second tables with the server address used and a corresponding proxy identifier of the proxy that serviced the request.

23. The apparatus as recited in claim 17, further comprising a portion of the proxy configured to receive the requested information from the other proxy, the retrieved information having been retrieved by the other proxy, and to forward the retrieved information to the client.

24. The apparatus as recited in claim 17, further comprising a portion of the other proxy configured to retrieve by the other proxy the requested information and to forward by the other proxy the retrieved information to the client.

25. An apparatus that accesses information in a network, comprising:

means in a proxy for receiving a request for access to a network resource stored on a server, wherein the request includes a network resource address corresponding to the network resource;

means for determining an address of the server from the network resource address;

means for comparing the server address against a plurality of server addresses stored in a first table of the proxy, wherein each stored address corresponds to a server to which the proxy has a persistent connection;

means for, if a match for the server address cannot be found in the first table, interrogating a second table of the proxy, which contains a list of at least one other proxy and corresponding stored addresses to which respective ones of the other proxies have a persistent connection, to find an address that matches the server address; and

means for transferring the request for access, if an address stored in the second table matches the server address, to another proxy that corresponds to the address found in the second table.

26. An apparatus that accesses information in a network, comprising:

a portion of a proxy configured to receive a request for access to a network resource stored on a server, wherein the request includes a network resource address corresponding to the network resource;

a portion of the proxy configured to determine an address of the server from the network resource address;

a portion of the proxy configured to compare the server address against a plurality of server addresses stored in a first table of the proxy, wherein each stored address corresponds to a server to which the proxy has a persistent connection;

a portion of the proxy configured to broadcast, by the proxy, if a match for the server address cannot be found in the first table, a request for service to other proxies listed in a second table of the proxy;

a portion of the proxy configured to receive, by the proxy, an indication from at least one other proxy that a persistent connection exists between the other proxy and the server; and

a portion of the proxy configured to transfer, if a persistent connection exists between the other proxy and the server, the request for access to the other proxy.

27. The apparatus as recited in claim 26, further comprising a portion of the proxy configured to retrieve, if the server address corresponds to an address stored in the first table of the proxy, information from the server at that address via an existing persistent connection between the proxy and the server.

28. The apparatus as recited in claim 27, further comprising a portion of the proxy configured to send the information retrieved from the server to the client.

29. The apparatus as recited in claim 28, further comprising a portion of the proxy configured to, when a persistent connection is terminated, update the first and second tables to indicate that the persistent connection no longer exists.

30. The apparatus as recited by claim 26, further comprising a portion of the proxy configured to, if the broadcast request does not yield a response from any of the listed proxies and a timeout occurs, initiate a persistent connection to the server and to retrieve the requested information.

31. The apparatus as recited in claim 30, further comprising a portion of the proxy configured to forward the retrieved information to the client.

32. The apparatus as recited in claim 26, further comprising a portion of the proxy configured to receive the requested information that has bee retrieved by the other proxy and to forward the retrieved information to the client.

33. The apparatus as recited in claim 26, further comprising a portion of the other proxy configured to retrieve the requested information and to forward by the other proxy the retrieved information to the client.

34. An apparatus that accesses information in a network, comprising:

means in a proxy for receiving a request by a proxy for access to a network resource stored on a server, wherein the request includes a network resource address corresponding to the network resource;

means for determining an address of the server from the network resource address;

means for comparing the server address against a plurality of server addresses stored in a first table of the proxy, wherein each stored address corresponds to a server to which the proxy has a persistent connection;

means for broadcasting, by the proxy, if a match for the server address cannot be found in the first table, a request for service to other proxies listed in a second table of the proxy;

means for receiving, by the proxy, an indication from at least one other proxy that a persistent connection exists between the other proxy and the server; and

means for transferring, if a persistent connection exists between the other proxy and the server, the request for access to the other proxy.

35. A computer program product comprising:

a computer useable medium having computer readable code embodied therein for causing accessing of information in a network, the computer program product including:

computer program code devices configured to receive by a proxy a request for access to a network resource stored on a server, wherein the request includes a network resource address corresponding to the network resource;

computer program code devices configured to determine an address of the server from the network resource address;

computer program code devices configured to compare the server address against a plurality of server addresses stored in a first table of the proxy, wherein each stored address corresponds to a server to which the proxy has a persistent connection;

computer program code devices configured to, if a match for the server address cannot be found in the first table, interrogate a second table of the proxy, which contains a list of other proxies and corresponding stored addresses to which respective ones of the other proxies have a persistent connection, to find an address that matches the server address; and

computer program code devices configured to transfer the request for access, if an address stored in the second table matches the server address, to another proxy that corresponds to the address found in the second table.

36. A computer program product comprising:

a computer useable medium having computer readable code embodied therein for causing accessing of information by a proxy in a network, the computer program product including:

computer program code devices configured to receive a request for access to a network resource stored on a server, wherein the request includes a network resource address corresponding to the network resource;

computer program code devices configured to determine an address of the server from the network resource address;

computer program code devices configured to compare the server address against a plurality of server addresses stored in a first table of the proxy, wherein each stored address corresponds to a server to which the proxy has a persistent connection;

computer program code devices configured to broadcast, if a match for the server address cannot be found in the first table, a request for service to other proxies listed in a proxy support table of the proxy;

computer program code devices configured to receive an indication from at least one other proxies that a persistent connection exists between the other proxy and the server; and

computer program code devices configured to transfer, if a persistent connection exists between the other proxy and the server, the request for access to the other proxy.

37. A computer data signal embodied in a carrier wave and representing sequences of instructions which, when executed by the processor, causes said processor to access information in a client/server network, by performing the steps of:

receiving a request for access to a network resource stored on a server, wherein the request includes a network resource address corresponding to the network resource;

determining an address of the server from the network resource address;

comparing the server address against a plurality of server addresses stored in a first table of the proxy, wherein each stored address corresponds to a server to which the proxy has a persistent connection;

if a match for the server address cannot be found in the first table, interrogating a second table of the proxy, which contains a list of other proxies and corresponding stored addresses to which respective ones of the other proxies have a persistent connection, to find an address that matches the server address; and

transferring the request for access, if an address stored in the second table matches the server address, to another proxy that corresponds to the address found in the second table.

38. A computer data signal embodied in a carrier wave and representing sequences of instructions which, when executed by the processor, causes said processor to access information in a client/server network, by performing the steps of:

receiving a request for access to a network resource stored on a server, wherein the request includes a network resource address corresponding to the network resource;

determining an address of the server from the network resource address;

comparing the server address against a plurality of server addresses stored in a first table of the proxy, wherein each stored address corresponds to a server to which the proxy has a persistent connection;

broadcasting, if a match for the server address cannot be found in the first table, a request for service to other proxies listed in a proxy support table of the proxy;

receiving an indication from at least one other proxies that a persistent connection exists between the other proxy and the server; and

transferring, if a persistent connection exists between the other proxy and the server, the request for access to the other proxy.

of the proxy that serviced the request.

wherein executing a first computer program to forward the information received from the network resource provider to the client.

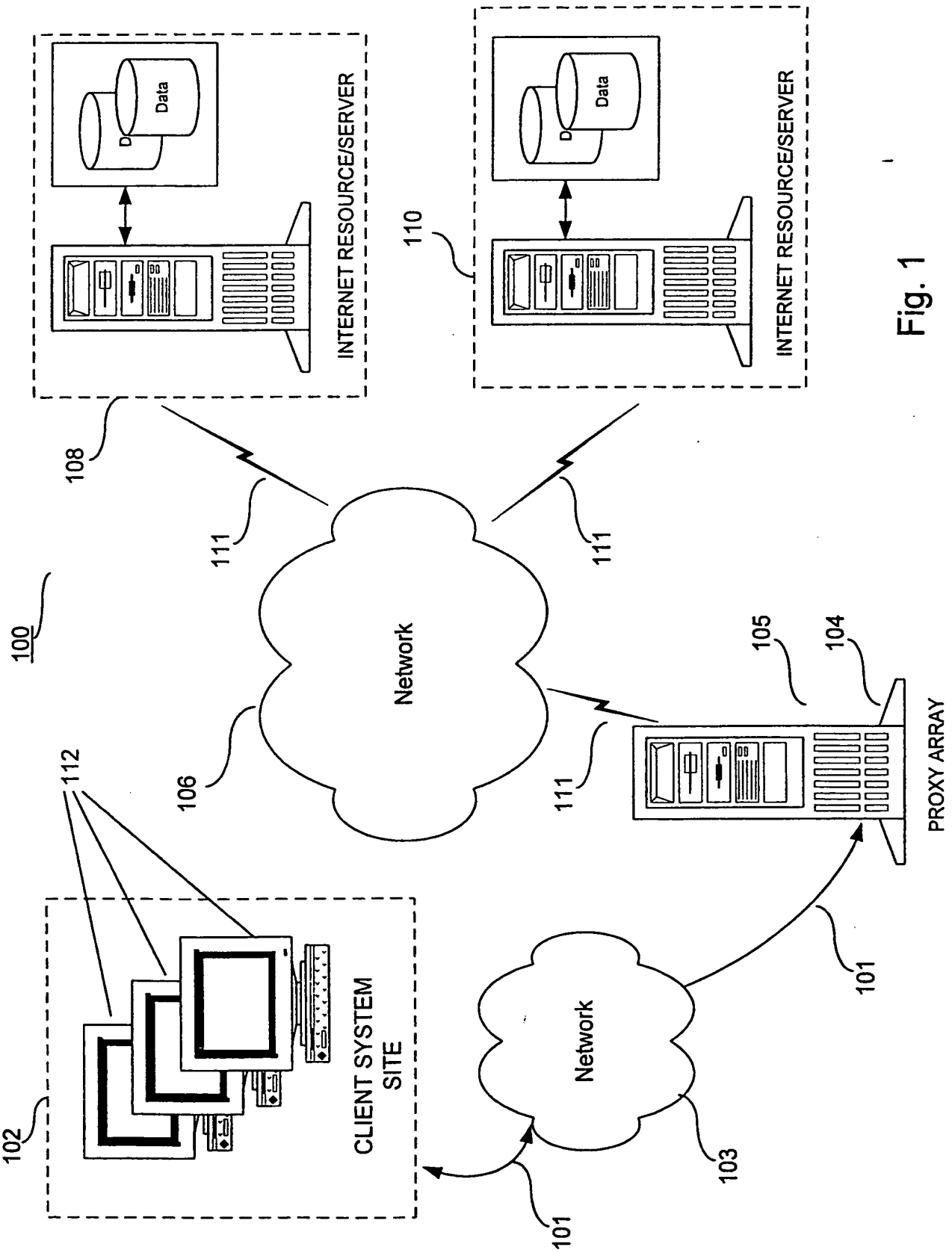

Data
INTERNET RESOURCE/SERVER
108
110
111
111
100
Network
106
105
104
111
PROXY ARRAY
112
CLIENT SYSTEM SITE
102
101
Network
103
101

Fig. 1

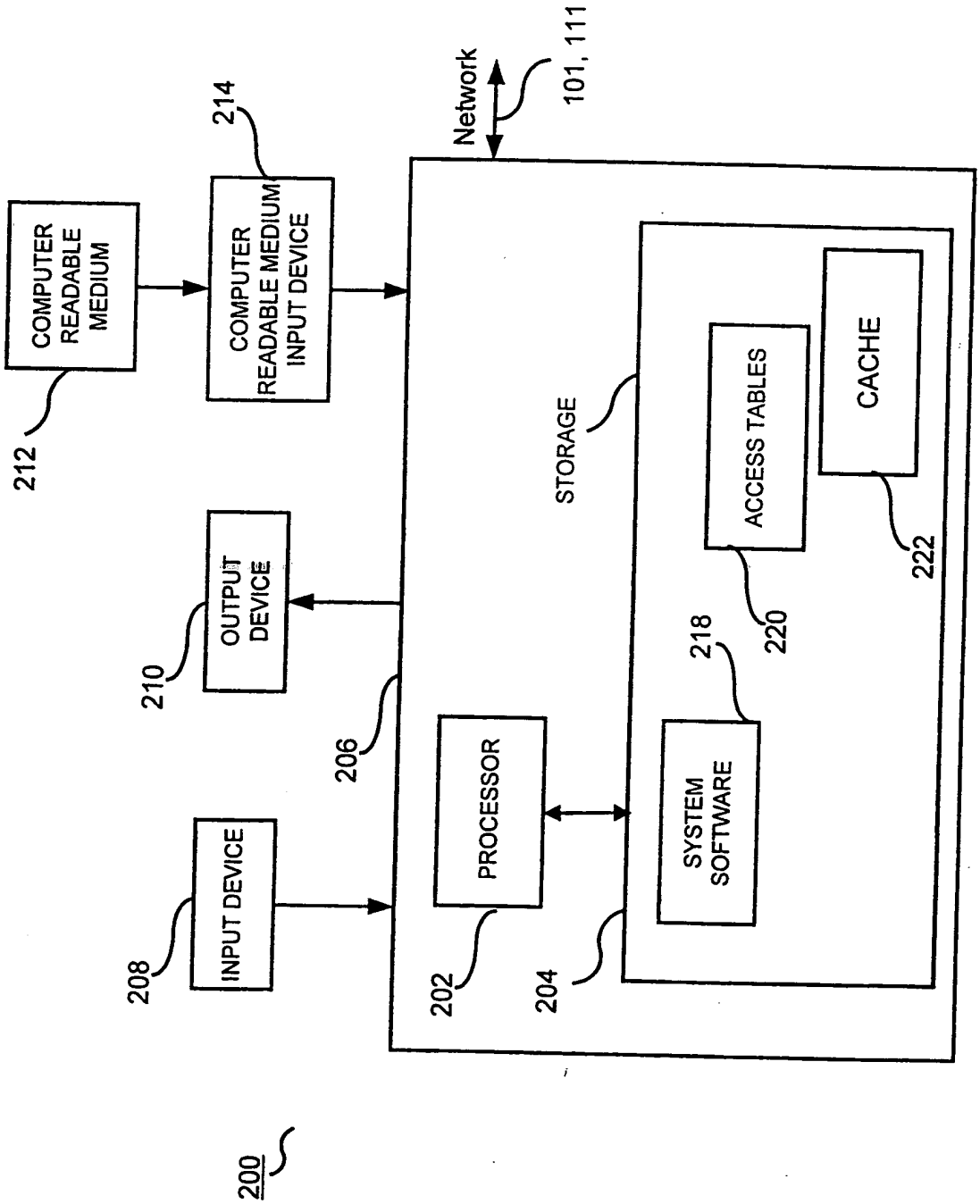

Network
101, 111
214
COMPUTER READABLE MEDIUM
COMPUTER READABLE MEDIUM INPUT DEVICE
212
STORAGE
ACCESS TABLES
CACHE
222
220
218
OUTPUT DEVICE
210
206
PROCESSOR
SYSTEM SOFTWARE
INPUT DEVICE
208
202
204
200

Fig. 2
Proxy

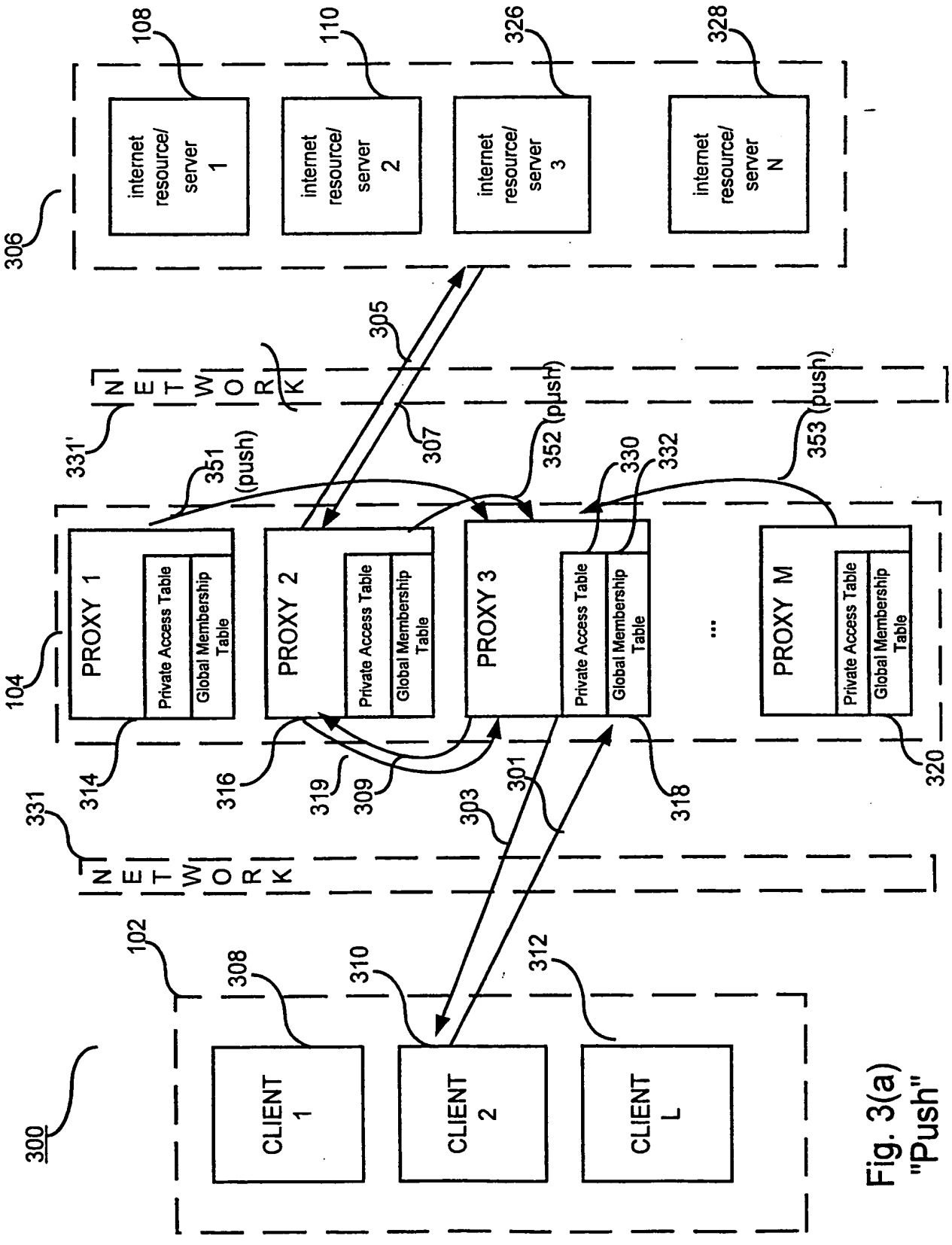

300
102
CLIENT 1
CLIENT 2
CLIENT L
308
310
312
331
NETWORK
104
PROXY 1
Private Access Table
Global Membership Table
PROXY 2
PROXY 3
PROXY M
314
316
319
309
303
301
318
320
330
332
331'
351 (push)
352 (push)
353 (push)
307
305
306
108
110
326
328
internet resource/ server 1
internet resource/ server 2
internet resource/ server 3
internet resource/ server N

Fig. 3(a)
"Push"

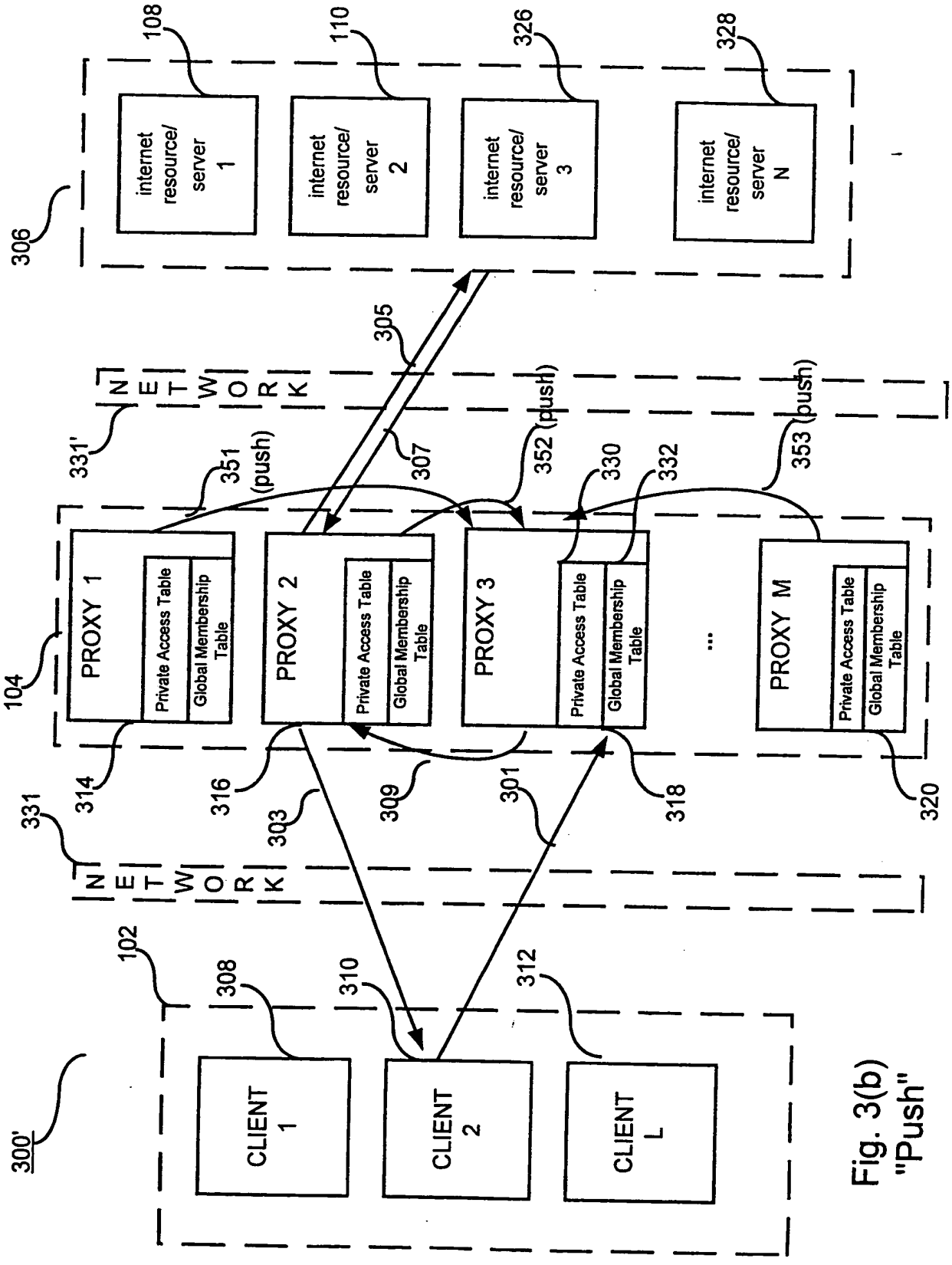

300'
102
CLIENT 1
CLIENT 2
CLIENT L
308
310
312
331
NETWORK
314
316
303
309
301
318
320
104
PROXY 1
Private Access Table
Global Membership Table
PROXY 2
Private Access Table
Global Membership Table
PROXY 3
Private Access Table
Global Membership Table
...
PROXY M
Private Access Table
Global Membership Table
331'
NETWORK
351 (push)
307
305
352 (push)
330
332
353 (push)
306
internet resource/ server 1
internet resource/ server 2
internet resource/ server 3
internet resource/ server N
108
110
326
328

Fig. 3(b)
"Push"

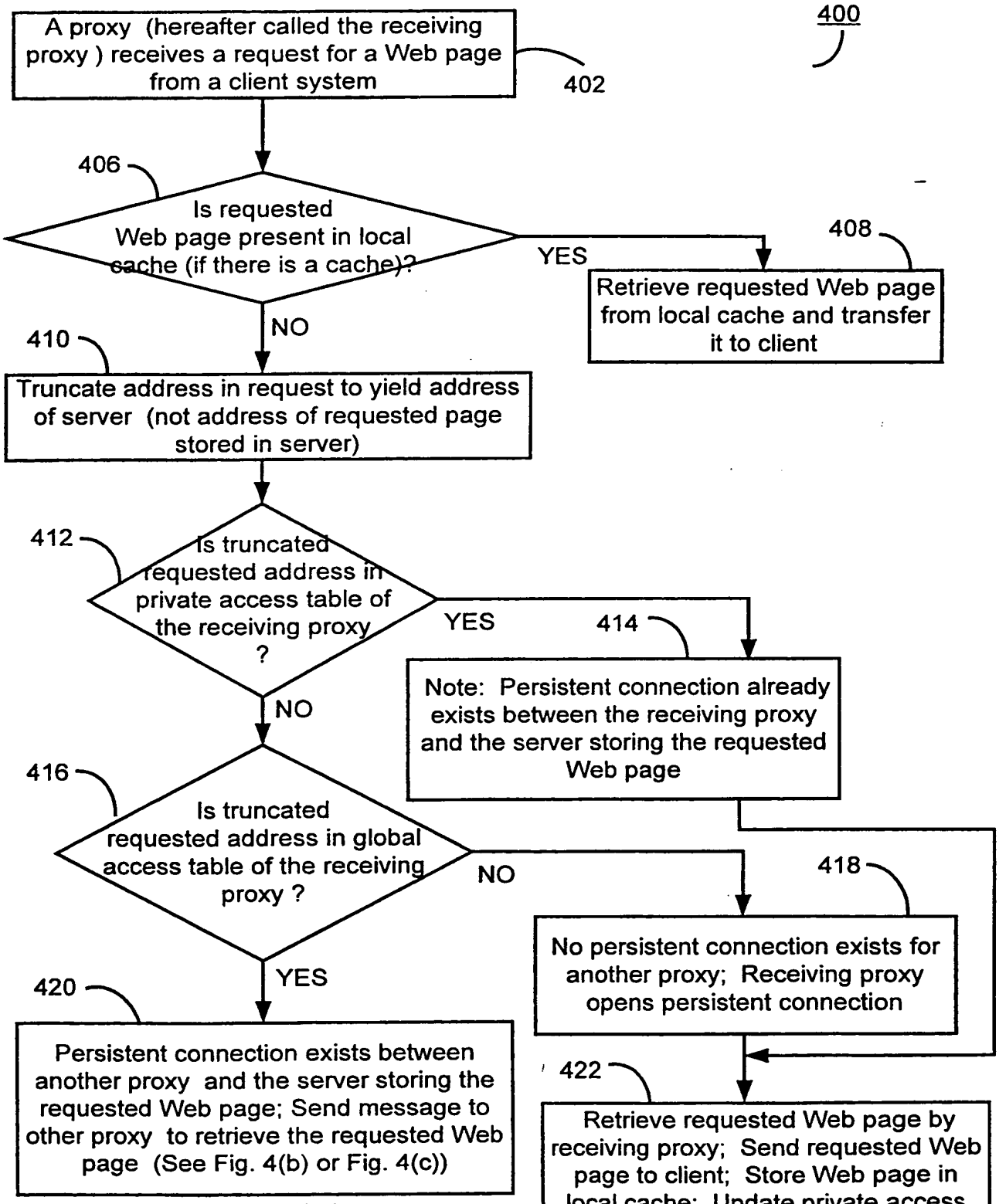


Fig. 4(a)
Receiving a request for a web page by a receiving proxy

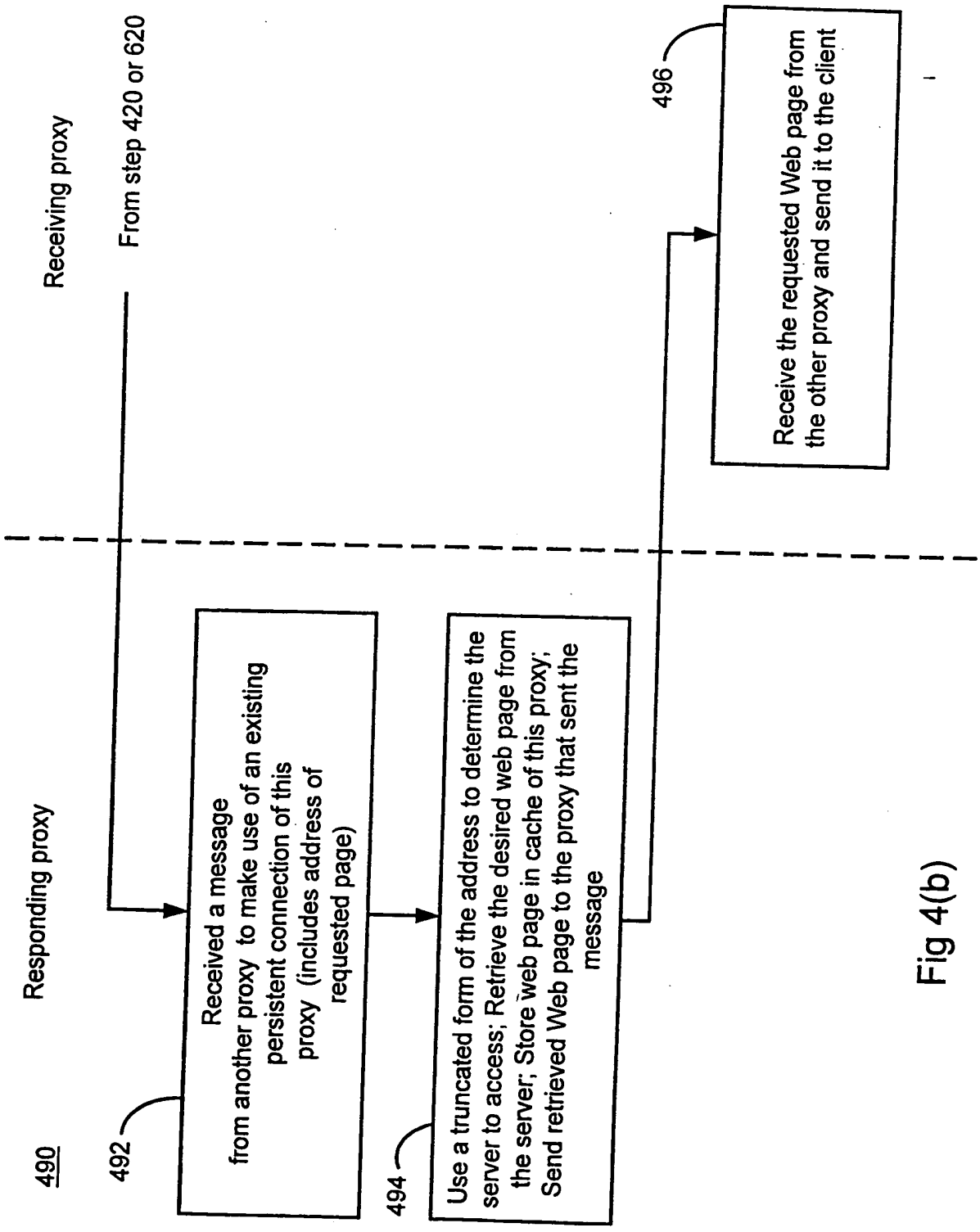

490
Responding proxy
Receiving proxy
From step 420 or 620
492
Received a message from another proxy to make use of an existing persistent connection of this proxy (includes address of requested page)
494
Use a truncated form of the address to determine the server to access; Retrieve the desired web page from the server; Store web page in cache of this proxy; Send retrieved Web page to the proxy that sent the message
496
Receive the requested Web page from the other proxy and send it to the client

Fig 4(b)

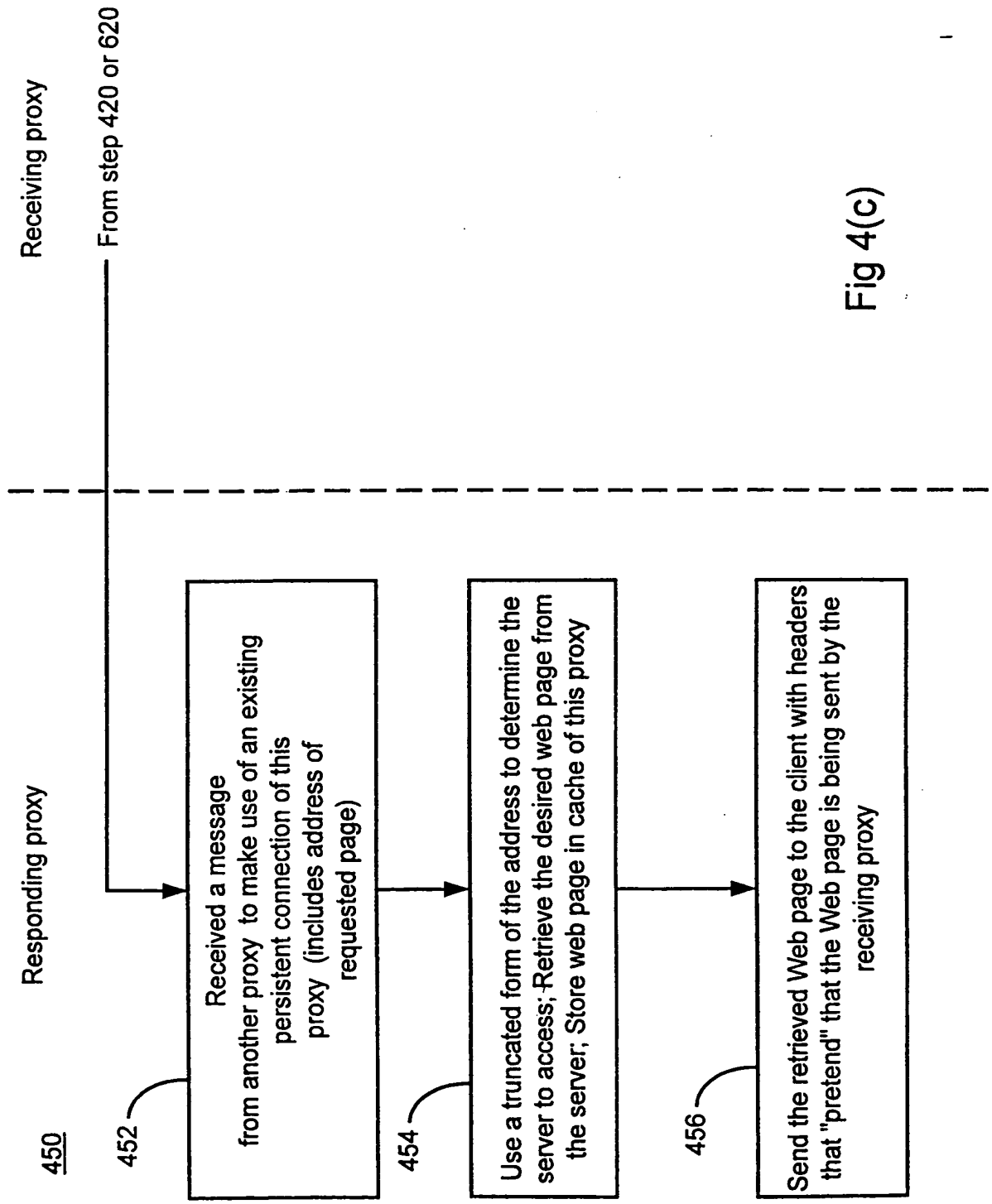

Receiving proxy
From step 420 or 620
Responding proxy
450
452
Received a message from another proxy to make use of an existing persistent connection of this proxy (includes address of requested page)
454
Use a truncated form of the address to determine the server to access; Retrieve the desired web page from the server; Store web page in cache of this proxy
456
Send the retrieved Web page to the client with headers that "pretend" that the Web page is being sent by the receiving proxy

Fig 4(c)

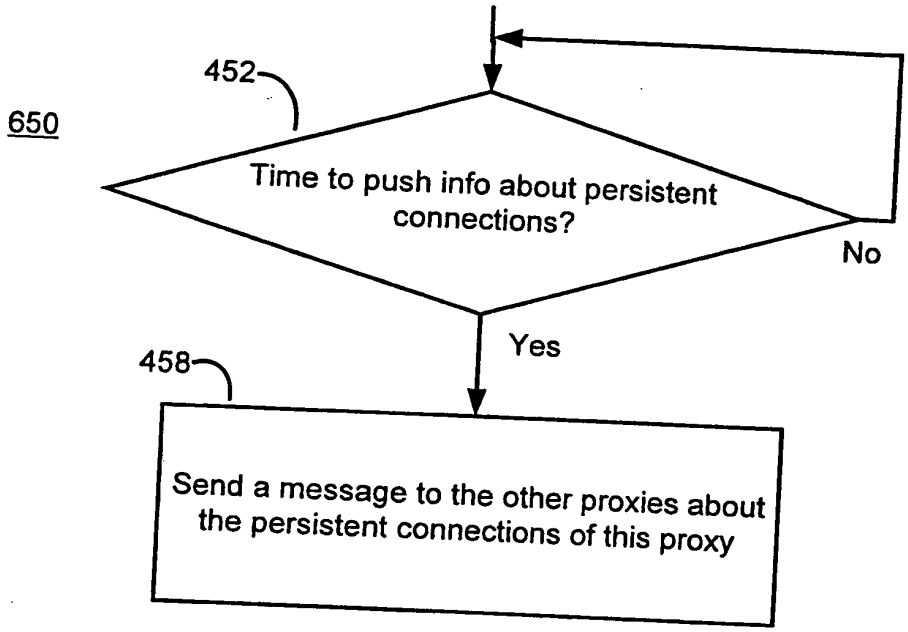


Fig 4(d)
Pushing persistent
connection information

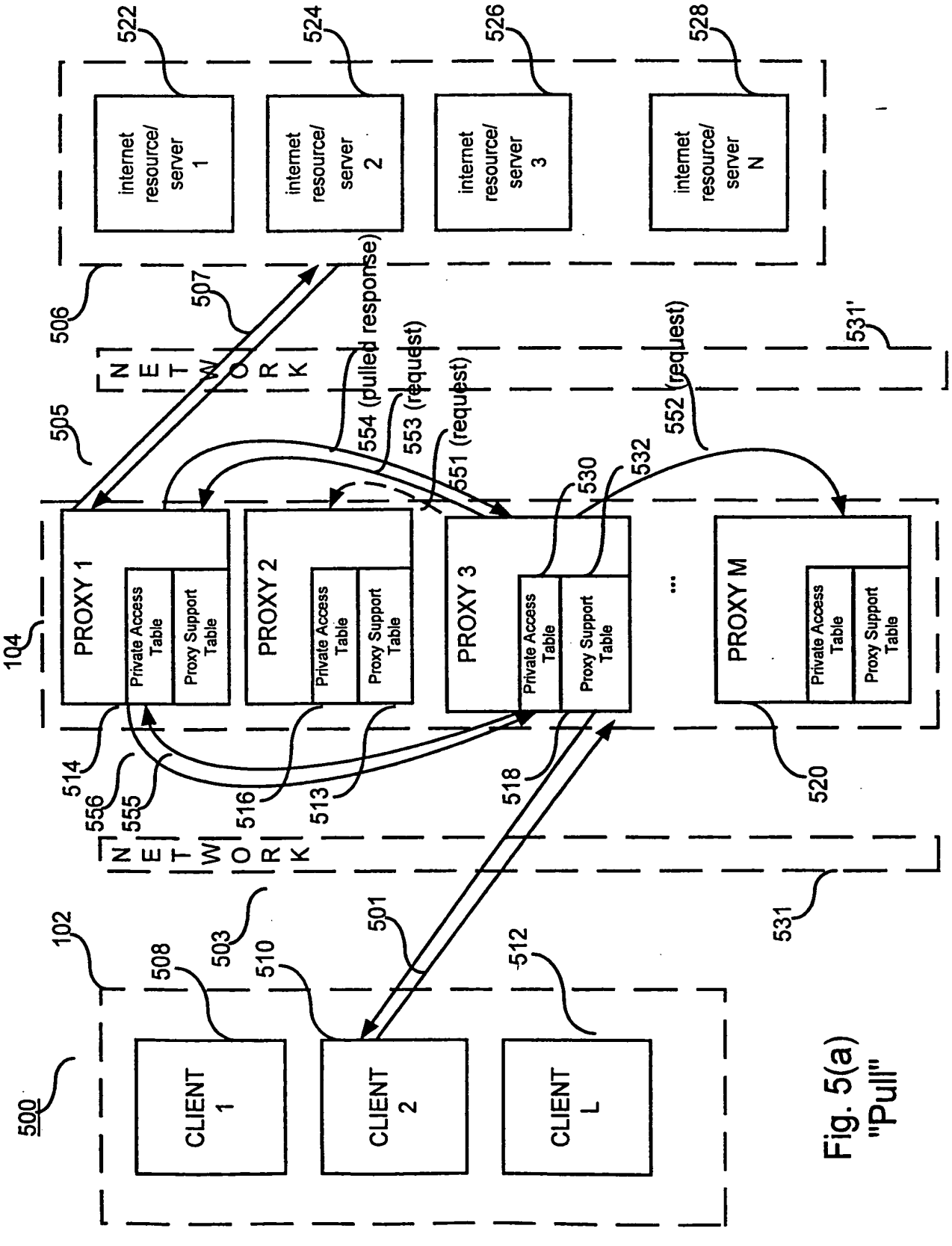

internet resource/ server 1
internet resource/ server 2
internet resource/ server 3
internet resource/ server N
522
524
526
528
506
507
505
NETWORK
531'
554 (pulled response)
553 (request)
551 (request)
552 (request)
530
532
PROXY 1
PROXY 2
PROXY 3
PROXY M
Private Access Table
Proxy Support Table
104
514
556
555
516
513
518
520
531
102
503
501
508
510
512
500
CLIENT 1
CLIENT 2
CLIENT L

Fig. 5(a)
"Pull"

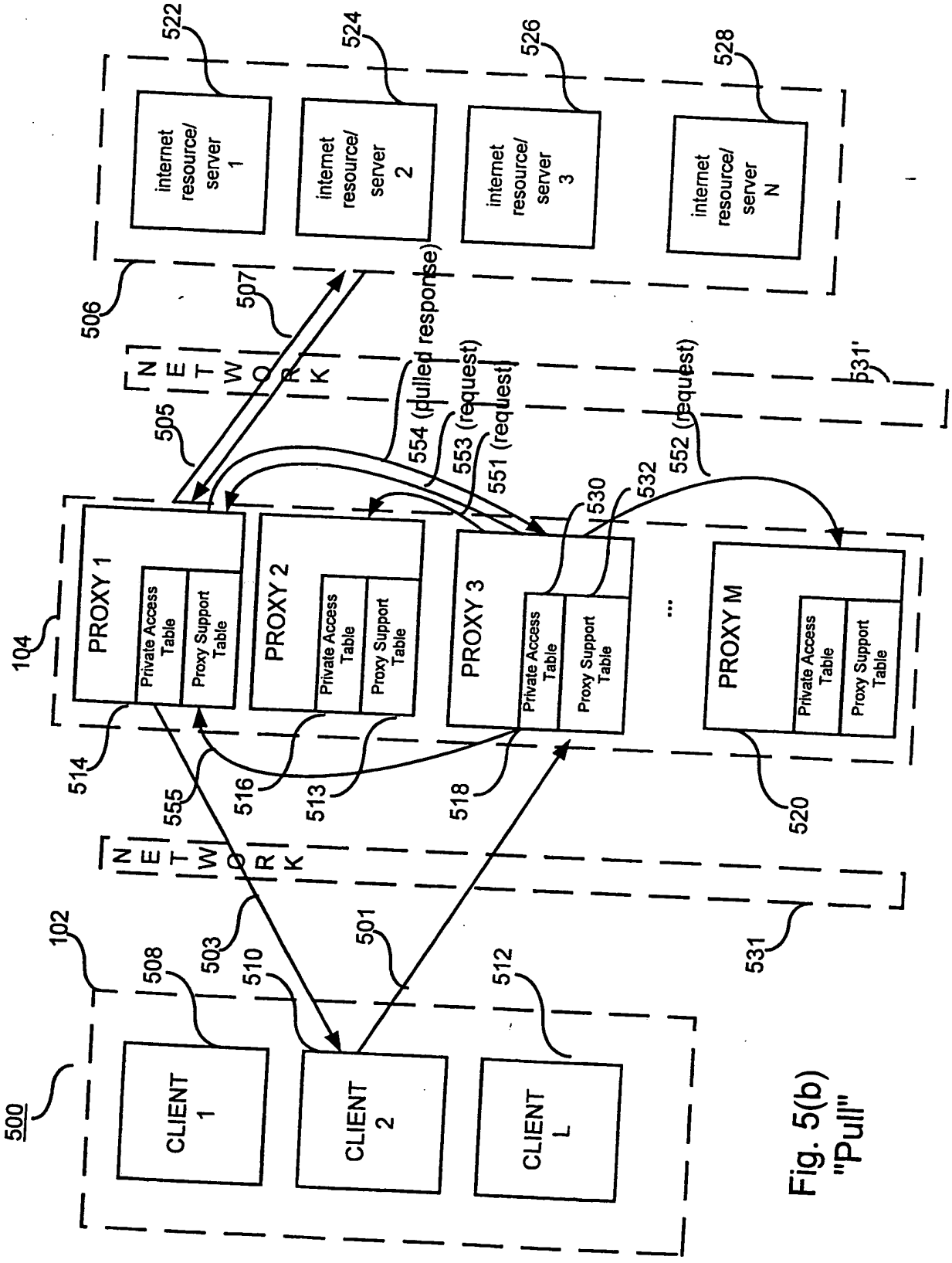

522
524
526
528
internet resource/ server 1
internet resource/ server 2
internet resource/ server 3
internet resource/ server N
506
507
NETWORK
505
554 (pulled response)
553 (request)
551 (request)
552 (request)
531'
530
532
PROXY 1
Private Access Table
Proxy Support Table
PROXY 2
PROXY 3
PROXY M
104
514
555
516
513
518
520
NETWORK
531
102
503
501
508
510
512
500
CLIENT 1
CLIENT 2
CLIENT L

Fig. 5(b)
"Pull"

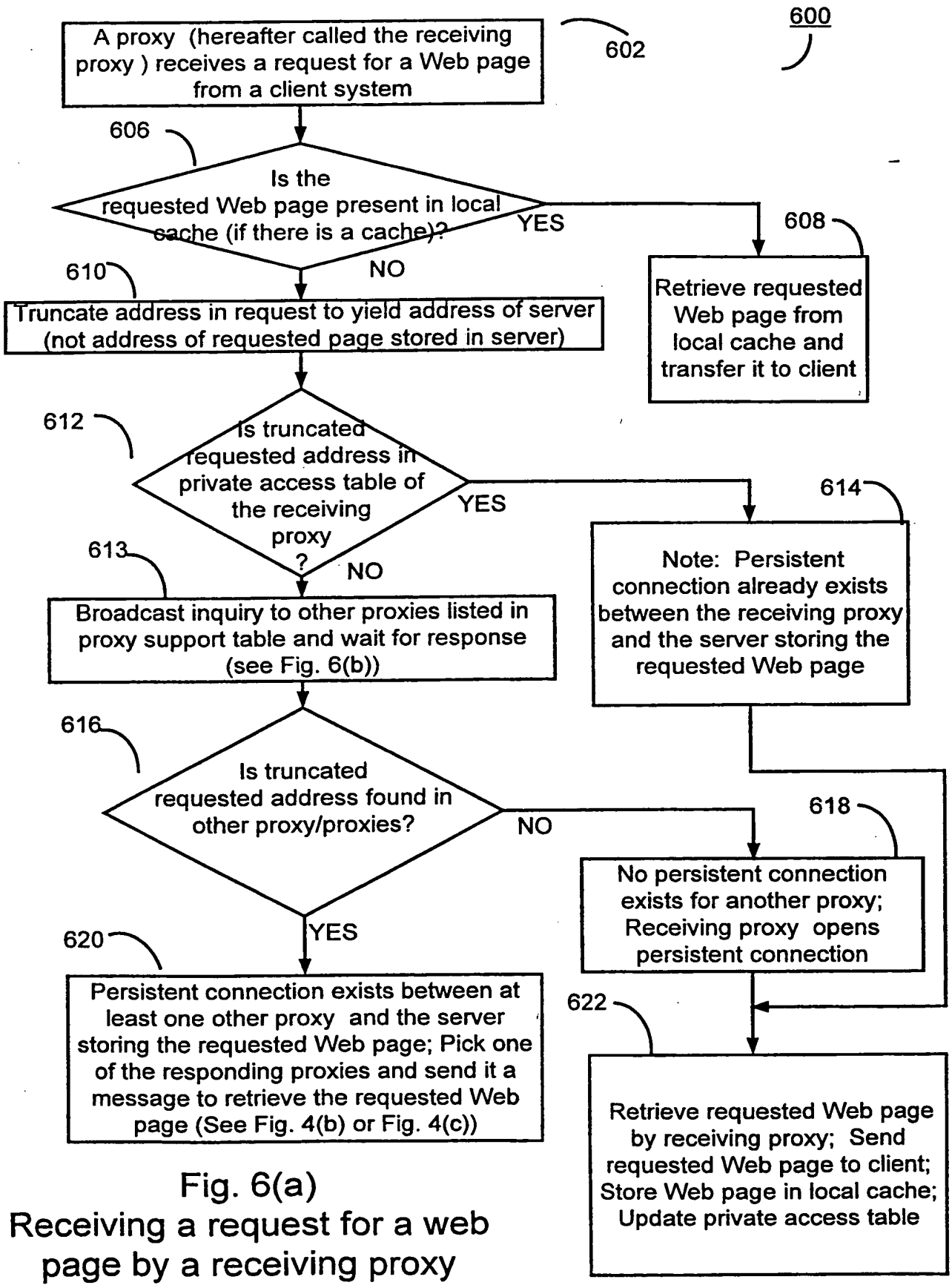


Fig. 6(a)
Receiving a request for a web page by a receiving proxy

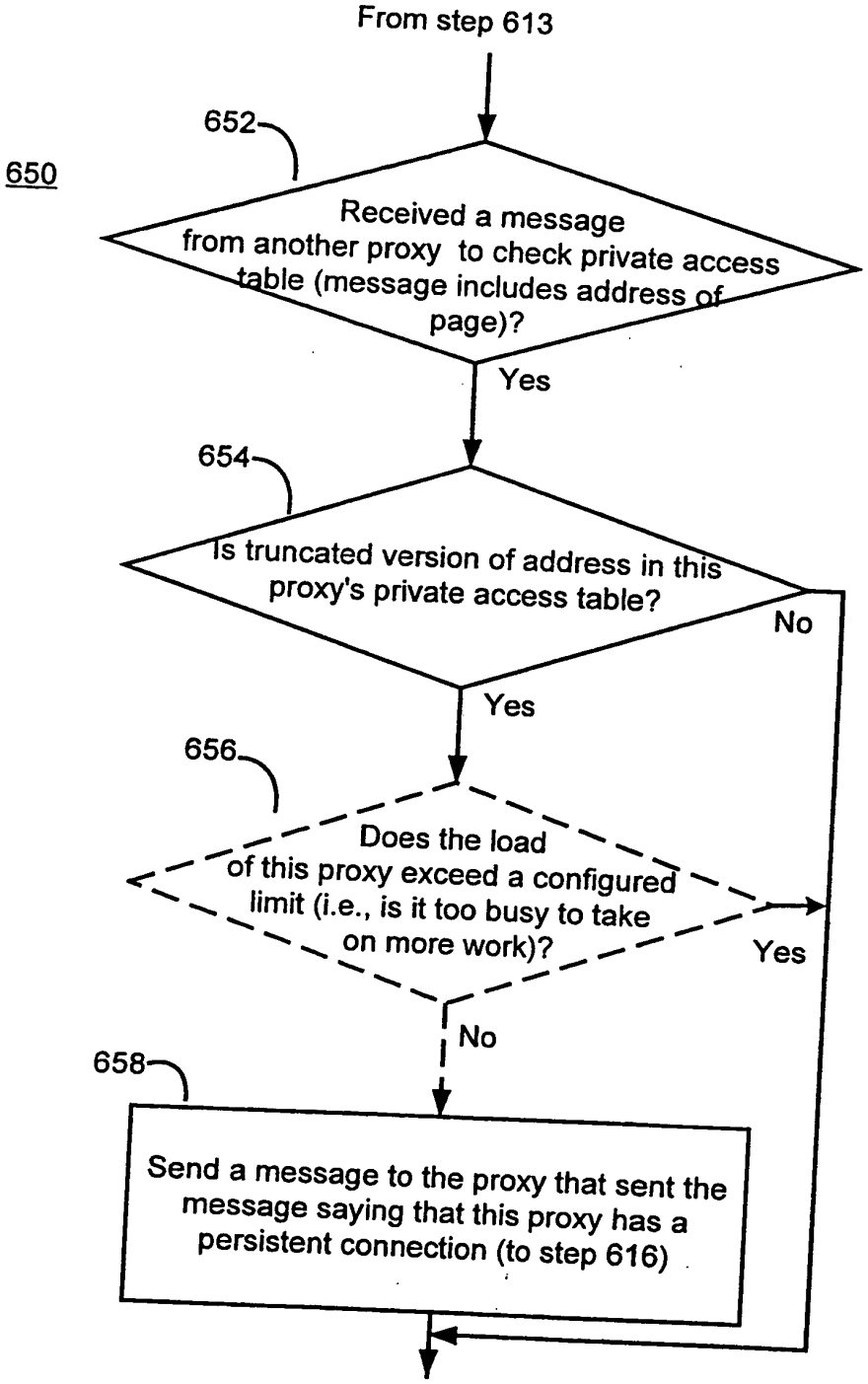


Fig 6(b)
Pulling persistent connection information

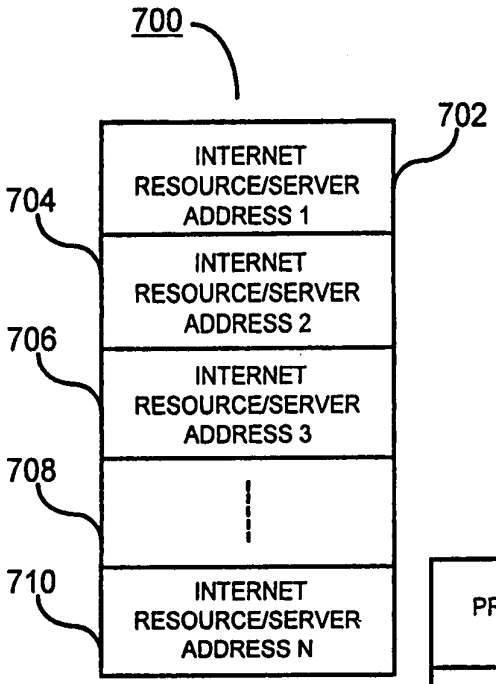


Fig. 7
Private Access Table Format
(The proxy containing this table has a persistent connection to these resources/ servers)

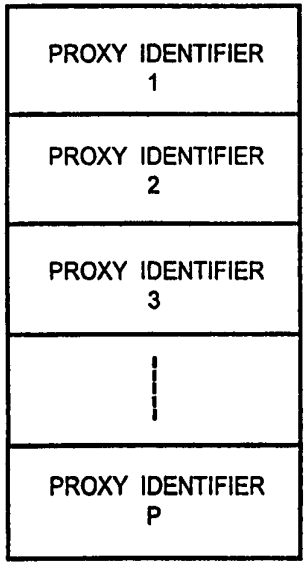


Fig. 9
Proxy Support Table Format
(Ask these proxies whether they have a persistent connection)

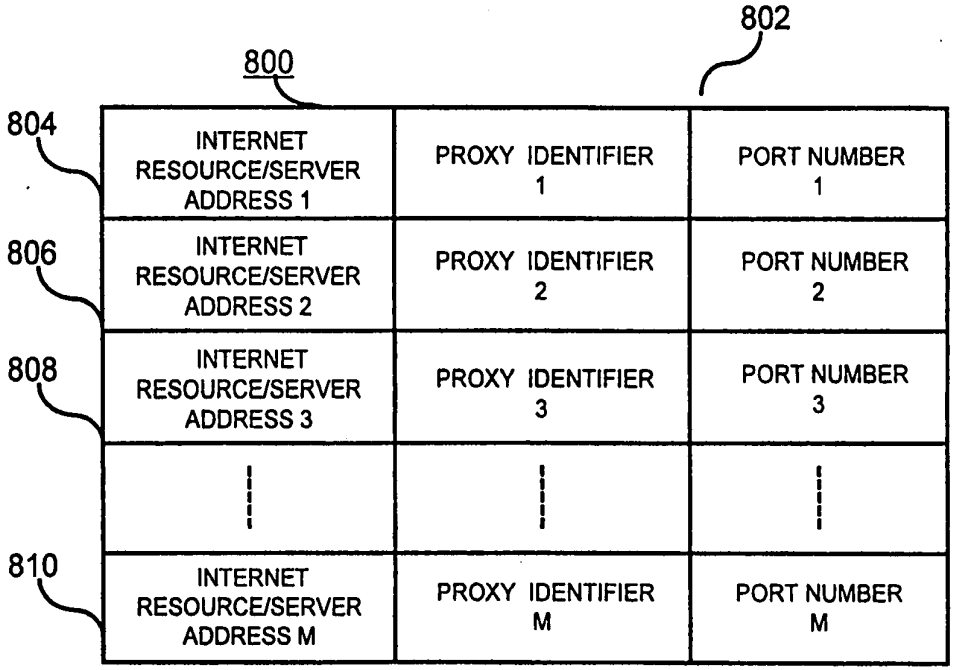


Fig. 8
Global Membership Table Format
(These proxies have a persistent connection with these resources)

INTERNATIONAL SEARCH REPORT

onal Application No
PCT/US 99/20720

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06 H04L29/12

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	WO 98 17039 A (LINDBO SVERKER ;LOETHBERG PETER (SE); MIRROR IMAGE INTERNET AB (SE) 23 April 1998 (1998-04-23) page 3, line 17 -page 8, line 1 page 8, line 24-33 page 11, line 8 -page 12, line 12 page 13, line 1 -page 15, line 2 page 15, line 20 -page 16, line 18 figure 3 --- -/--	1-3,8, 17-19, 25,35,37 4-7, 9-16, 20-24, 26-34, 36,38

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
7 February 2000	17/02/2000
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Lázaro Lõpez, M.L.

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 99/20720

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication.where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 828 367 A (MATSUSHITA ELECTRIC IND CO LTD) 11 March 1998 (1998-03-11) column 1, line 48 -column 3, line 31 column 3, line 46 -column 4, line 28 ---	1-38
A	US 5 774 660 A (LIU ZAIDE ET AL) 30 June 1998 (1998-06-30) column 2, line 41-67 column 6, line 41 -column 7, line 30 column 10, line 38 -column 12, line 63 ---	1-38
A	EP 0 865 180 A (LUCENT TECHNOLOGIES INC) 16 September 1998 (1998-09-16) column 4, line 38 -column 6, line 12 column 8, line 35 -column 9, line 28 column 10, line 29-51 -----	1-38

INTERNATIONAL SEARCH REPORT

Information on patent family members

onal Application No
PCT/US 99/20720

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9817039 A	23-04-1998	SE 507138 C	06-04-1998
		AU 4640797 A	11-05-1998
		EP 0966820 A	29-12-1999
		SE 9603753 A	06-04-1998
EP 0828367 A	11-03-1998	JP 10065737 A	06-03-1998
		CN 1175031 A	04-03-1998
US 5774660 A	30-06-1998	NONE	
EP 0865180 A	16-09-1998	CA 2230550 A	14-09-1998